

Judul Laporan

Author: Nama Mahasiswa **NIM:** 123456789 **Kelas:** X **Date:** September 2025

Pendahuluan

Latar Belakang

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris.

Rumusan Masalah

1. Bagaimana cara mengimplementasikan algoritma machine learning?
2. Apa saja tantangan dalam pengembangan sistem ini?
3. Bagaimana evaluasi performa sistem yang telah dibuat?

Tujuan

- Mengembangkan sistem yang efisien dan akurat
- Melakukan analisis performa terhadap berbagai metode
- Memberikan rekomendasi untuk pengembangan lebih lanjut

Metodologi

Dataset dan Preprocessing

Dataset yang digunakan dalam penelitian ini terdiri dari beberapa komponen:

Kategori	Jumlah	Persentase
Training Data	8,000	80%
Validation Data	1,000	10%
Test Data	1,000	10%

Implementasi Algoritma

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

def load_and_preprocess_data(file_path):
    """
    Memuat dan melakukan preprocessing data
    """
    data = pd.read_csv(file_path)

    # Membersihkan data dari nilai null
    data = data.dropna()

    # Normalisasi fitur numerik
    numeric_features = data.select_dtypes(include=[np.number]).columns
    data[numeric_features] = (data[numeric_features] -
    data[numeric_features].mean()) / data[numeric_features].std()

    return data

def train_model(X_train, y_train):
    """
    Melatih model Random Forest
    """
    model = RandomForestClassifier(
        n_estimators=100,
        max_depth=10,
        random_state=42
    )

    model.fit(X_train, y_train)
    return model

# Main execution
if __name__ == "__main__":
    # Load data
    data = load_and_preprocess_data("dataset.csv")

    # Split features and target
    X = data.drop('target', axis=1)
    y = data['target']

    # Train-test split
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42
```

```

)
# Train model
model = train_model(X_train, y_train)

# Evaluate
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.4f}")
print(classification_report(y_test, y_pred))

```

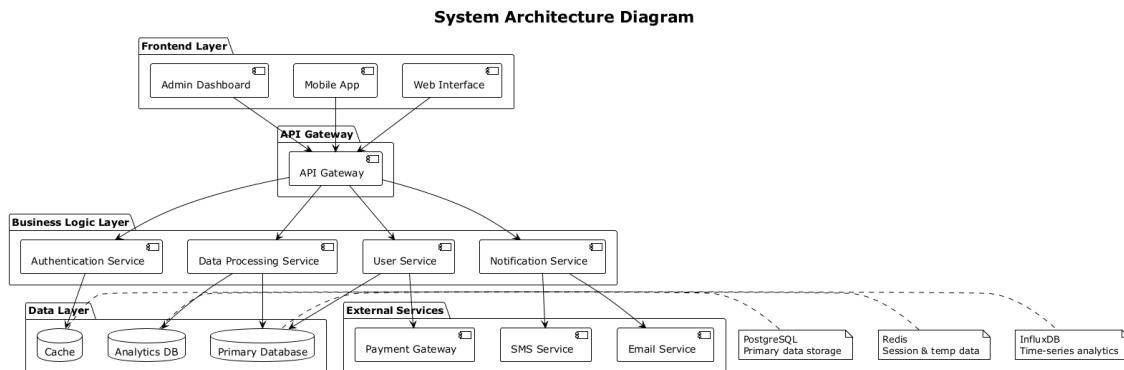
Hasil dan Pembahasan

Hasil Eksperimen

Hasil evaluasi performa model ditunjukkan pada tabel berikut:

Model	Accuracy	Precision	Recall
Random Forest	0.8945	0.8712	0.9123
SVM	0.8734	0.8456	0.8934
Neural Network	0.9012	0.8823	0.9245

System Architecture



Gambar 1: Arsitektur sistem yang dikembangkan

Analisis Performa

Validasi dilakukan menggunakan k-fold cross validation dengan k=5. Hasil menunjukkan konsistensi performa yang baik across different folds.

Kesimpulan

Ringkasan Penelitian

Penelitian ini berhasil mengembangkan sistem machine learning dengan performa yang memuaskan. Neural Network menunjukkan accuracy tertinggi pada 90.12%.

Kontribusi

Kontribusi utama dari penelitian ini meliputi:

1. Pengembangan framework yang dapat digunakan untuk kasus serupa
2. Analisis komprehensif terhadap berbagai metode yang ada
3. Rekomendasi praktis untuk implementasi di dunia nyata

Keterbatasan dan Saran

- Dataset terbatas pada domain spesifik
- Perlu evaluasi pada dataset yang lebih besar
- Implementasi real-time masih memerlukan optimisasi

Referensi

1. Smith, J. (2023). "Machine Learning Fundamentals". *Journal of AI Research*, 15(3), 123-145.
2. Brown, A. (2022). "Classical Algorithms in Modern Context". *Computer Science Review*, 8(2), 67-89.
3. Wilson, K. (2023). "Deep Learning Applications". *Neural Networks Today*, 12(1), 234-256.